Truncation-Induced Dependencies among Summary Statistics (TIDES)

Ian Hussey & Jamie Cummins

Abstract

Measures used to collect data often place constraints on the data they generate, including truncation (or bounding) and granularity. For example, Likert scales, which are commonly used in the social sciences. These constraints induce dependencies among summary statistics generated from such data, such as Mean, Standard Deviation and the number of observations (N). ...

Likert scales are very common the social sciences, as is the reporting of summary statistics such as mean (μ), Standard Deviation (SD) and number of observations (n). The nature of the measure used to collect the data places several constraints on and dependencies among the summary statistics that may be observed. These constraints and dependencies are useful to establish as they have utility in the detection of inconsistent or impossible results reported in published articles that may have arisen due to scientific errors or research integrity issues.

For example, a single item 1-7 Likert scale generates data is *truncated* (or *bounded*) in the sense that scores below 1 or above 7 are not possible to observe. We can say that scale generates data with the lower and upper bounds $[\ell, u]$. Of course, other values may find their way into a dataset and erroneously be used to calculate summary statistics for other reasons, such as the scoring of missing data with a placeholder value such as 99.

Additionally, a single item 1-7 Likert scale also generates data is *granular* in the sense that scores must be integers (1 to 7, but not decimals such as 1.3). Again, strange exceptions can exist where other values are introduced, such as a participant who circles both "2" and "3" on the questionnaire and has their score recorded as "2.5", but such special cases are relatively rare.

Another obvious constraint is that the mean score must also lie within the range of the scale, such that

$$n\ell \leq n\mu \leq nu. \tag{1}$$

This dependency among the scale's bounds and the data's μ has the benefit of often being possible to assess without access to the dataset itself, as these bounds are often reported in the methods section (e.g., "a single item 1-7 Likert scale") and the mean is often reported in the results. This is particularly useful given that data for publications is frequently not available (see REF).

Additional dependencies as a function of the constraints of truncation and bounding exist, but these become increasingly less obvious to the untrained eye. First, the fact that such a Likert scale's individual observations have a known granularity also induces dependencies between the n and mean score: only some means are possible for a given sample size and vice-versa. To take the simplest example, if there are only two observations, the mean of those observations must end in .0 or .5. Conversely, a mean such as 0.66 is incompatible with an N of 2. The observation is referred to as the Granularity-Related Inconsistency of Means (GRIM: REF).

Second, comparable dependencies exist between the sample size and the Standard Deviation. An intuitive example is not easy to provide but, by analogy, just as not all means are possible for a given n, it is also the case that not all Standard Deviations are possible for a given n. This observation is referred to as

GRIMMER (REF). GRIM and GRIMMER have demonstrated utility in the assessment of inconsistent and impossible results in scientific articles, many of which are relatively benign (but nonetheless erroneous, and therefore distort results), but in some cases were used to uncover research integrity violations (REFs).

GRIM, and to a lesser extent GRIMMER, are useful because they can be applied to results reported in articles without access to the raw data, they are relatively simple, and there are several accessible implementations of them (REFs). However, both suffer from the fact that their utility is limited due to the fact that means and Standard Deviations are typically rounded when reported in articles. For mathematical reasons, this means that GRIM tests will always be passed when the mean is reported to two decimal places and the $n >= 100$ (see REF for discussion - NOTE THAT IT'S TECHNICALLY N_PARticipants*n_items IN FOOTNOTE). There are therefore practical reasons to explore what additional dependencies exist among summary statistics such as mean, Standard Deviation and n that are induced by constraints such as truncation and granularity.

This article builds on previous work on the max sd... error focus ... local sd/mean dependency .. sprite.

Maximum SD of a truncated variable

Problem statement

This section addresses the problem of finding the maximum Standard Deviation for a dataset with fixed sample size n, desired mean μ, and observations constrained to integers within a specified range $[\ell, u]$. Under these constraints, what is the smallest possible Standard Deviation that can be observed?

Given n observations $x_1, x_2, \ldots, x_n$, each constrained to the interval $[\alpha, \beta]$ (where α = min_score and β = max_score), we want to determine the distribution of these observations that yields the maximum sample standard deviation, subject to the mean (μ), which itself lies in range $[\ell, \beta]$.

A closed form solution for maximizing variance under constraints

For real numbers in $[\alpha, \beta]$, the arrangement that maximizes variance (and therefore Standard Deviation) is to place as many observations as possible at the boundaries α and β. Intuitively, a distribution with more "spread" around its mean exhibits higher variance.

TODO ADD THE PLOTS ILLUSTRATING THIS HERE - I HAVE CODE FOR THIS ELSEWHERE.

Concretely, assume p observations are at β and q observations are at α. If $p + q = n$ exactly satisfies the sum constraint $p\,\beta + q\,\alpha = n\,\mu$, then all values lie at the extremes. However, in many cases, one additional "middle" value m in $[\alpha, \beta]$ is required to adjust the total sum to exactly $n\,\mu$. In such scenarios:

- p observations are at β.
- $(n - p - 1)$ observations are at α.
- Exactly one observation is at $m = \alpha + \delta$, for some $\delta \in (0,\ \beta - \alpha)$.

Determining the number of maximum values

To find how many observations should be at β, we solve

$$p\beta + (n - p)\,\alpha \leq n\,\mu, \tag{4}$$

which implies

$$p \leq \frac{n\,\mu - n\,\alpha}{\beta - \alpha}. \tag{5}$$

Hence, the largest integer p we can use is

$$p = \left\lfloor \frac{n\,\mu - n\,\alpha}{\beta - \alpha} \right\rfloor. \tag{6}$$

Calculating the leftover δ

After setting p values to β and $n - p$ values to α, the current sum is

$$p\,\beta + (n - p)\,\alpha = n\,\alpha + p\,(\beta - \alpha). \tag{7}$$

The difference between the required sum $n\,\mu$ and the sum so far is

$$\delta = n\,\mu - \big[p\,\beta + (n - p)\,\alpha\big]. \tag{8}$$

Since δ must fit within $[\alpha, \beta]$ for a single observation, it can be shown that that $\delta \in [\,0,\, \beta - \alpha)$. If $\delta = 0$, then no middle value is needed. If $\delta > 0$, one observation is set to $\alpha + \delta$.

Showing $\delta \in [0, \beta - \alpha)$

By substituting the expression for the sum so far into the equation for δ:

$$\delta = n\mu - \big[n\alpha + p(\beta - \alpha)\big]. \tag{9}$$

Expanding this gives:

$$\delta = n\mu - n\alpha - p(\beta - \alpha). \tag{10}$$

Rearranging terms:

$$\delta = n(\mu - \alpha) - p(\beta - \alpha). \tag{11}$$

Lower Bound for δ: The minimum value of δ occurs when p is maximized, that is, $p = \left\lfloor \frac{n(\mu-\alpha)}{\beta-\alpha} \right\rfloor$. Substituting this into δ, we see that the leftover sum always satisfies $\delta \geq 0$, since the sum cannot exceed the required total.

Upper Bound for δ: The maximum value of δ occurs when one observation must adjust to make up the exact total. In this case, δ corresponds to the leftover amount of the total that does not perfectly divide into $p\beta + (n-p)\alpha$, and since all values lie within $[\alpha, \beta]$, the leftover is strictly less than $\beta - \alpha$. Thus, $\delta \in [0, \beta - \alpha)$.

If $\delta = 0$, no middle value is needed because the sum perfectly divides into $p\beta$ and $(n - p)\alpha$. Otherwise, one observation is assigned $\alpha + \delta$.

Variance computation

Let $\overline{x} = \mu$. The sum of squared deviations from the mean is:

- p observations at β:
$$p \times (\beta - \mu)^2. \tag{12}$$
- $(n - p - 1)$ observations at α (if $\delta > 0$) or $(n - p)$ if $\delta = 0$:
$$(n - p - 1) \times (\alpha - \mu)^2 \quad \text{or} \quad (n - p) \times (\alpha - \mu)^2. \tag{13}$$
- One "middle" observation (only if $\delta > 0$):
$$(\alpha + \delta - \mu)^2. \tag{14}$$

Summing these yields the total sum of squares SS:

$$\text{SS} = p \times (\beta - \mu)^2 + (n - p - 1) \times (\alpha - \mu)^2 + (\alpha + \delta - \mu)^2 \quad \text{if } \delta > 0, \tag{15}$$

or

$$\text{SS} = p \times (\beta - \mu)^2 + (n - p) \times (\alpha - \mu)^2 \quad \text{if } \delta = 0. \tag{16}$$

The sample variance is

$$s^2 = \frac{\text{SS}}{n - 1}, \tag{17}$$

and the sample Standard Deviation is $s = \sqrt{s^2}$.

Because the distribution is "as extreme as possible," this arrangement yields a larger SS (and hence s^2) than any other arrangement that still satisfies $\sum x_i = n\,\mu$. Thus, one obtains the maximum possible sample standard deviation (or variance) under the stated constraints.

Additional considerations for practical use

- SDs reported in articles are rounded or truncated for reporting. Assessment of $SD_{reported} \leq SD_{max}$ should take various possibilities into account and consider their precision.
- Accommodate multi-item scales.

Implementation in R

This method can be implemented in R as follows:

```r
max_sd <- function(n, min_score, max_score, mean, integer_responses = TRUE) {
  # 1) required sum
  required_sum <- n * mean

  # check: is required_sum in feasible range [n*min_score, n*max_score]?
  if (required_sum < n*min_score - 1e-9 || required_sum > n*max_score + 1e-9) {
    stop("No integer distribution can achieve this mean with responses in [min_score,max_score].")
  }
```

```
  # round required_sum if the problem implies the mean is exactly
  # representable by integer responses:
  if(integer_responses){
    required_sum <- round(required_sum)
  }

  # 2) solve for x
  x <- floor((required_sum - n*min_score) / (max_score - min_score))
  x <- max(0, min(x, n))  # keep it in [0, n]

  # 3) leftover = required_sum - [x*max_score + (n-x)*min_score]
  delta <- required_sum - (x*max_score + (n - x)*min_score)
  if (delta < 0 || delta >= (max_score - min_score)) {
    stop("Something went wrong with leftover. Check input.")
  }

  # 4) sum of squares around mean:

  # part A: from x responses at max_score
  ss_max_score <- x * (max_score - mean)^2

  # part B: from (n - x - 1) responses at min_score (if we do have a leftover)
  # or from (n - x) responses at min_score (if no leftover)
  if (delta == 0) {
    # no middle value
    ss_min_score <- (n - x) * (min_score - mean)^2
    ss_m <- 0
  } else {
    ss_min_score <- (n - x - 1) * (min_score - mean)^2
    middle_val <- min_score + delta
    ss_m <- (middle_val - mean)^2
  }

  # total sum of squares
  ss_total <- ss_max_score + ss_min_score + ss_m

  # sample variance = ss_total / (n - 1)
  var_max <- ss_total / (n - 1)
  sd_max  <- sqrt(var_max)

  data.frame(max_sd = sd_max)
}
```

The Figure 1 illustrates the use of this function to calculate the maximum SD for each possible value of the mean, in increments of .01, for a variable that is truncated [1,7], for N = 12. Two output plots are provided, one where the variable is continuous but truncated and one where it granular (e.g., a Likert scale where responses are integers).

```
# dependencies
library(dplyr)
library(tidyr)
library(purrr)
library(ggplot2)
library(scales)
```

```r
# calculations
res <-
  expand_grid(n = 12,
              min_score = 1,
              max_score = 7,
              mean = seq(1, 7, by = 0.01),
              integer_responses = c(TRUE, FALSE)) |>
  mutate(result = pmap(list(n, min_score, max_score, mean, integer_responses),
                       possibly(max_sd, otherwise = NA))) |>
  unnest(result) |>
  mutate(label = case_when(integer_responses == TRUE ~
                             "Bounded granular response variable (Likert)",
                           integer_responses == FALSE ~
                             "Bounded continuous response variable"))

# plot
ggplot(data = res, aes(mean, max_sd)) +
  geom_line() +
  scale_x_continuous(breaks = breaks_pretty(n = 9)) +
  theme_linedraw() +
  theme(panel.grid.minor = element_blank()) +
  facet_wrap(~ label, ncol = 1) +
  ylim(0, 3.25) +
  xlab("Mean") +
  ylab("Max SD")
```

Minimum SD of a truncated variable

Problem statement

This section addresses the problem of finding the minimum Standard Deviation for a dataset with fixed sample size n, desired mean μ, and observations constrained to integers within a specified range $[\ell, u]$. Under these constraints, what is the smallest possible Standard Deviation that can be observed?

A closed form solution for minimising variance under constraints

Suppose we have a sample size n, a mean μ, and know that observations are constrained to be within the interval $[\ell, u]$, where ℓ and u are the minimum and maximum possible integer scores, respectively (e.g., Likert scale points).

We wish to minimize the sample standard deviation under these constraints. The key result is that, if it is feasible to have all observations equal to some integer $x \in [\ell, u]$ that matches the required mean, then the standard deviation is 0, which is obviously minimal. Formally, if $n\mu$ is an integer and $\ell \leq \mu \leq u$ (with μ itself an integer), one can set $X_1 = \cdots = X_n = \mu$, yielding:

$$s^2 = 0 \tag{18}$$

However, if μ is not an integer or if μ is outside the feasible range of a single integer value, the best strategy is to distribute observations across two adjacent integers. Concretely, let

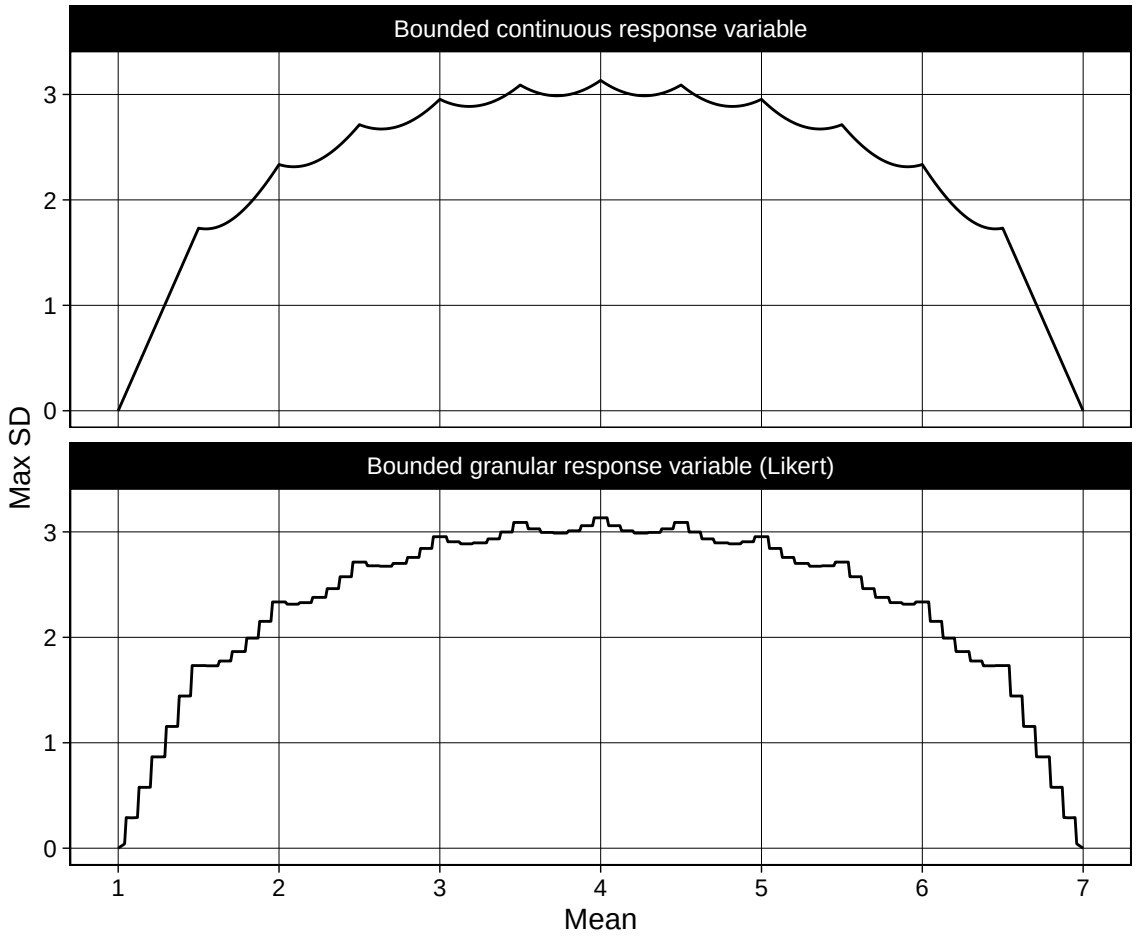

Figure 1: Maximum SD for a given N across means (increments of 0.1)

$$L = \max\{\ell,\ \lfloor \mu \rfloor\}, \tag{19}$$

be the largest integer not exceeding μ (clamped to be at least ℓ), and let $r = n\mu - nL$ be the "remainder" when forcing the mean to come from a baseline value L. In a purely theoretical (unbounded) setting, distributing r fraction of the observations at $L+1$ and the remainder at L yields the exact mean μ. Specifically, $n-r$ observations are placed at L and r observations are placed at $L+1$.

Since the scale is integer-valued, r must be an integer. If r falls between 0 and n and $\ell \le L \le u$ and $L+1 \le u$, then the distribution

$$(L, L, \ldots, L, \underbrace{L+1, \ldots, L+1}_{r \text{ times}}), \tag{20}$$

matches the sum $n\mu$ and satisfies the bounds.

No other arrangement of integer-valued observations with the same sum can yield a smaller variance, because any additional "spreading out" away from these two points (L and $L+1$) can only increase the sum of squared deviations from the mean.

Variance computation

If we take $n-r$ observations at L and r observations at $L+1$, then the sample mean is forced to be $\mu \approx L + r/n$.

The population variance of this two-point distribution can be expressed as:

$$s^2_{\text{pop}} = (1-p)(L-\mu)^2 + p((L+1)-\mu)^2, \tag{21}$$

where $p = \frac{r}{n}$ is the proportion of observations at $L+1$.

We expand and simplify this to:

$$s^2_{\text{pop}} = \frac{r}{n}((L+1)-\mu)^2 + \left(1 - \frac{r}{n}\right)(L-\mu)^2. \tag{22}$$

Expanding further and substituting $\mu = L + \frac{r}{n}$:

$$s^2_{\text{pop}} = \frac{r}{n}\left(\frac{n-r}{n}\right)^2 + \frac{n-r}{n}\left(-\frac{r}{n}\right)^2. \tag{23}$$

The sample variance (with denominator $n-1$) is:

$$s^2 = \frac{n}{n-1} \cdot s^2_{\text{pop}}. \tag{24}$$

and the sample Standard Deviation is $s = \sqrt{s^2}$.

By substituting these values, we arrive at the closed-form solution for the minimal Standard Deviation. This approach confirms that the minimal-variance distribution uses at most two adjacent integer values, balanced according to the required mean.

Summary: The smallest-variance distribution for integer constraints and a given mean uses at most two adjacent integer values, with exact weights determined by the remainder $r = n\mu - nL$.

Implementation in R

This method can be implemented in R as follows:

```r
min_sd <- function(n, min_score, max_score, mean, integer_responses = TRUE) {
  # ---- 1) required total sum (rounded to an integer if the problem implies
  # exact feasibility)
  required_sum <- n * mean

  # round required_sum if the problem implies the mean is exactly representable
  # by integer responses:
  if(integer_responses){
    required_sum <- round(required_sum)
  }

  # feasibility check: the sum must lie between n*min_score and n*max_score
  # for an integer distribution to exist.
  if (required_sum < n*min_score || required_sum > n*max_score) {
    stop("No distribution of [min_score, max_score] integers can achieve that mean (sum out of range).")
  }

  # ---- 2) check if the mean is effectively an integer within [min_score,max_score].
  # if so, the minimal SD is 0 by taking all responses = that integer.
  if (abs(required_sum - n*round(mean)) < 1e-9 && round(mean) >= min_score && round(mean) <= max_score) {
    # i.e. if mean was effectively an integer in the feasible range
    # check if n * round(mean) == required_sum
    if (round(mean) * n == required_sum) {
      # all responses equal to that integer
      dist <- rep(round(mean), n)
      return(data.frame(min_sd = 0))
    }
  }

  # ---- 3) otherwise, we attempt to use two adjacent integers L and L+1
  # let L = floor(mean), clamped to [min_score, max_score].
  L_init <- floor(mean)
  if (L_init < min_score) L_init <- min_score
  if (L_init > max_score) L_init <- max_score

  # we'll define a small helper to build a distribution given L
  # and return if feasible:
  build_dist_from_L <- function(L, required_sum, n, min_score, max_score) {
    if (L < min_score || L > max_score) {
      return(NULL)
    }
    # leftover = how many times we need (L+1)
    leftover <- required_sum - n*L
    if (leftover == 0) {
      # all L
      return(rep(L, n))
    } else if (leftover > 0 && leftover <= n) {
      # leftover data points (L+1), the rest are L
      # but only if (L+1) <= max_score
      if ((L + 1) <= max_score) {
```

```
        return(c(rep(L, n - leftover), rep(L + 1, leftover)))
      } else {
        return(NULL)
      }
    } else {
      # leftover < 0 or leftover > n => not feasible with L and L+1
      return(NULL)
    }
  }

  # try L_init directly:
  dist <- build_dist_from_L(L_init, required_sum, n, min_score, max_score)
  if (is.null(dist)) {
    # if that didn't work, try adjusting L_init up or down by 1 step
    # (sometimes floor(mean) is not the right choice if leftover < 0 or > n).

    # let's define a small search around L_init:
    candidates <- unique(c(L_init - 1, L_init, L_init + 1))
    candidates <- candidates[candidates >= min_score & candidates <= max_score]

    found <- FALSE
    for (L_try in candidates) {
      dist_try <- build_dist_from_L(L_try, required_sum, n, min_score, max_score)
      if (!is.null(dist_try)) {
        dist <- dist_try
        found <- TRUE
        break
      }
    }

    if (!found) {
      # possibly no solution with 2 adjacent integers => no feasible distribution
      stop("Could not construct a minimal-variance distribution with two adjacent integers.")
    }
  }

  # if we reach here, 'dist' is a valid distribution
  # ---- 4) compute sample SD in R (with denominator n-1)
  min_sd <- sd(dist)

  data.frame(min_sd = min_sd)
}
```

Below I illustrate the use of this function to calculate the maximum SD for each possible value of the mean, in increments of .01, for a variable that is truncated [1,7], for N = 12 Two output plots are provided, one where the variable is continuous but truncated and one where it granular (e.g., a Likert scale where responses are integers).

```
# calculations
res <-
  expand_grid(n = 12,
              min_score = 1,
              max_score = 7,
              mean = seq(1, 7, by = 0.01),
```

```
              integer_responses = TRUE) |>
  mutate(result = pmap(list(n, min_score, max_score, mean, integer_responses),
                       possibly(min_sd, otherwise = NA))) |>
  unnest(result) |>
  mutate(label = case_when(integer_responses == TRUE ~
                             "Bounded granular response variable (Likert)"))

# plot
ggplot(data = res, aes(mean, min_sd)) +
  geom_line() +
  scale_x_continuous(breaks = breaks_pretty(n = 9)) +
  theme_linedraw() +
  theme(panel.grid.minor = element_blank()) +
  facet_wrap(~ label, ncol = 1) +
  ylim(0, 3.25) +
  xlab("Mean") +
  ylab("Min SD")
```

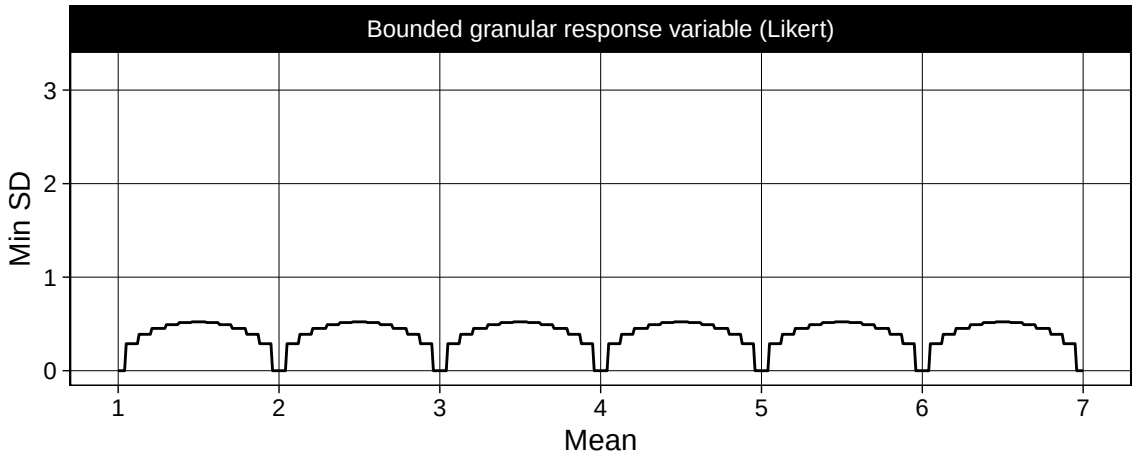

Figure 2: Minimum SD for a given N across means (increments of 0.1)

Visualising both minimum and maximum SDs

```
# calculations
res <-
  expand_grid(n = 12,
              min_score = 1,
              max_score = 7,
              mean = seq(1, 7, by = 0.01),
              integer_responses = c(TRUE, FALSE)) |>
  mutate(result = pmap(list(n, min_score, max_score, mean, integer_responses),
                       possibly(max_sd, otherwise = NA))) |>
  unnest(result) |>
  mutate(result = pmap(list(n, min_score, max_score, mean, integer_responses),
                       possibly(min_sd, otherwise = NA))) |>
  unnest(result) |>
```

```
  mutate(min_sd = ifelse(integer_responses, min_sd, NA),
         label = case_when(integer_responses == TRUE ~
                             "Bounded granular response variable (Likert)",
                           integer_responses == FALSE ~
                             "Bounded continuous response variable"))

# plot
ggplot(res) +
  geom_line(aes(mean, max_sd)) +
  geom_line(aes(mean, min_sd)) +
  scale_x_continuous(breaks = breaks_pretty(n = 9)) +
  theme_linedraw() +
  theme(panel.grid.minor = element_blank()) +
  facet_wrap(~ label, ncol = 1) +
  ylim(0, 3.25) +
  xlab("Mean") +
  ylab("Min SD")
```

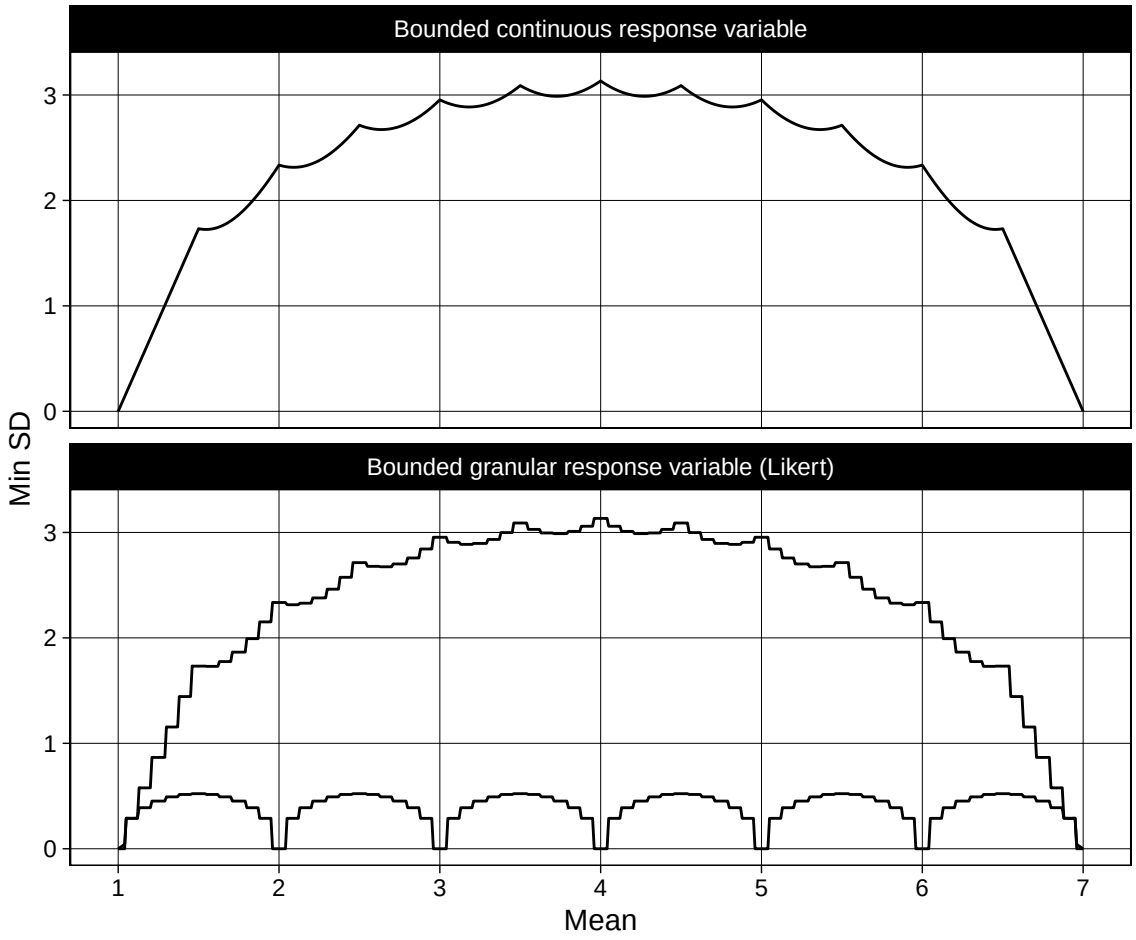

Figure 3: Minimum and maximum SD for a given N across means (increments of 0.1)

TODO

- Consider the choice of rounding method used inside the main function
- im less sure what integer_responses implies than i thought.
- ADD THE PLOTS ILLUSTRATING how max and min SD are determined HERE - I HAVE CODE FOR THIS ELSE-WHERE